MAP 512

Modèles pour le placement optimal pour les réseaux 5G

ÉCOLE POLYTECHNIQUE

Maya DELEGUE
Bianca MARIN MORENO

15 mars 2021

Table des matières

1 Introduction

Les réseaux 5G sont amenés à se développer rapidement dans les prochaines années. Ils devraient permettre de déployer de nombreux nouveaux services. En particulier, il est envisagé de piloter des véhicules autonomes grâce aux réseaux 5G. Cela nécessitera toutefois des délais de calcul et de transmission très courts. Les calculs ne pourront donc pas être effectués dans le cloud, mais uniquement via des serveurs de calcul situés à proximité des usagers, selon le principe du "Mobile Edge Computing" (MEC).

Déterminer où placer les serveurs afin de minimiser les délais tout en maîtrisant les contraintes de coût devient donc un enjeu majeur. Pour cela, on utilise des modèles de placement.

Ce problème est connu sous le nom de problème de Localisation d'entrepôt (*UFL : Uncapacited Facility Location*). Notre objectif est d'étudier un algorithme approximatif de ce problème proposé par l'article [1]. Cette approximation est basée sur le schéma Primal-Dual.

Pour ce faire, nous commencerons ce rapport par décrire le problème, sa relaxation et son dual, ainsi que ses conditions d'optimisation. Nous décrirons ensuite l'idée principale de l'algorithme d'approximation proposé dans l'article [1], quelques analyses, ainsi que les modifications que nous lui avons apportées lors du codage afin qu'il puisse fonctionner avec les instances que nous souhaitons. Nous décrirons l'algorithme à chaque passage dans un exemple réel, et nous terminerons avec le problème de k-médianes, aussi très importante et dont une solution approximative peut être trouvé en utilisant l'algorithme pour le problème de Localisation d'entrepôt. Toutes le comparaisons sont faites par-rapport à l'algorithme fournissant une solution exacte que nous avons aussi codé en Python.

2 Modélisation du problème

Le problème considéré est celui du placement d'installations. On considère un graphe G biparti, avec comme partition (F, C) où F représente l'ensemble des installations (*facilities* en anglais) et C celui des villes (*cities* en anglais). On note f_i le coût d'ouverture de l'installation i et c_{ij} le coût de connexion de la ville j à l'installation i. On recherche donc un sous-ensemble $I \subseteq F$ des installations qui vont être ouvertes, et une fonction $\Phi : C \to I$ qui attribue à chaque ville une installation ouverte en minimisant le coût total (somme des coûts d'ouverture des installations et de connexion des villes aux installations).

2.1 Programme Linéaire en Nombres Entiers

On peut donc modéliser notre problème par un programme linéaire en nombres entiers (PLNE). On note y_i la variable qui indique si l'installation i est ouverte ($y_i = 1$ si elle est ouverte, 0 sinon) et x_{ij} la variable qui indique si la ville j est connectée à l'installation i ($x_{ij} = 1$ si oui, 0 sinon).

Le problème s'écrit alors :

$$\min_{x_{ij}, y_i} \sum_{i \in F, j \in C} c_{ij} x_{ij} + \sum_{i \in F} f_i$$

sous contraintes :

$$\forall j \in C, \sum_{i \in F} x_{ij} \geq 1$$

$$\forall i \in F, \forall j \in C, y_i - x_{ij} \geq 0$$

$$\forall i \in F, \forall j \in C, x_{ij} \in \{0, 1\}$$

$$\forall i \in F, y_i \in \{0, 1\}$$

La première contrainte correspond au fait que chaque ville doit être reliée à au moins une installation. La seconde traduit le fait que l'installation à laquelle la ville est connectée est bien ouverte. Les deux dernières contraintes correspondent à la définition des variables binaires x_{ij} et y_i.

2.2 Relaxation linéaire du problème

La relaxation linéaire de ce PLNE consiste à ne plus demander que x_{ij} et y_i soient binaires. On arrive ainsi à un problème linéaire (PL) qui s'écrit alors :

$$\min_{x_{ij},y_i} \sum_{i\in F, j\in C} c_{ij}x_{ij} + \sum_{i\in F} f_i$$

sous contraintes :

$$\forall j \in C, \sum_{i\in F} x_{ij} \geq 1$$

$$\forall i \in F, \forall j \in C, y_i - x_{ij} \geq 0$$

$$\forall i \in F, \forall j \in C, x_{ij} \geq 0$$

$$\forall i \in F, y_i \geq 0$$

L'intérêt de faire une relaxation est que le PNLE est NP-dur, mais on sait résoudre efficacement les PL : par le simplexe (efficace en pratique), ou par les points intérieurs (efficace en pratique et en théorie), deux algorithmes étudiés pendant le cours de recherche opérationnelle [2]. Le rapport entre l'optimal dans la version relaxée et dans la version entière est souvent appelé *integrality gap*.

2.3 Problème dual

Pour calculer le dual il faut d'abord écrire le Lagrangien du problème.

On associe la variable duale α_j à la contrainte $1 - \sum_{i\in F} x_{ij} \leq 0$ et la variable duale β_{ij} à la contrainte $x_{ij} - y_i \leq 0$.

Le Lagrangien du problème s'écrit alors

$$L(x, y, \alpha, \beta) = \sum_{i\in F, j\in C} c_{ij}x_{ij} + \sum_{i\in F} f_i y_i + \sum_{j\in C} \alpha_j(1 - \sum_{i\in F} x_{ij}) + \sum_{i\in F, j\in C} \beta_{ij}(x_{ij} - y_i).$$

Le primal s'écrit alors

$$\inf_{x_{ij}\geq 0, y_i\geq 0} \sup_{\alpha_j\geq 0, \beta_{ij}\geq 0} L(x, y, \alpha, \beta).$$

Le dual s'écrit donc :

$$\sup_{\alpha_j\geq 0, \beta_{ij}\geq 0} \inf_{x_{ij}\geq 0, y_i\geq 0} L(x, y, \alpha, \beta).$$

On cherche à réécrire la partie $\inf_{x_{ij}\geq 0, y_i\geq 0} L(x, y, \alpha, \beta)$. On remarque qu'elle peut s'écrire en deux parties, une que l'on minimise en x et l'autre, indépendante de x, que l'on minimise en y uniquement. La première partie donne :

$$\inf_{x_{ij}\geq 0} \sum_{i\in F, j\in C} x_{ij}(c_{ij} - \alpha_j + \beta_{ij}).$$

Le minimum vaut 0 si $c_{ij} - \alpha_j + \beta_{ij} \geq 0$ et $-\infty$ sinon. La deuxième partie est :

$$\inf_{y_i\geq 0} \sum_{i\in F} y_i(f_i - \sum_{j\in C} \beta_{ij}).$$

Le minimum vaut 0 si $(f_i - \sum_{j\in C} \beta_{ij}) \geq 0$, et $-\infty$ sinon.

Cela revient donc à écrire le problème dual sous la forme suivante :

$$\max_{\alpha_j, \beta_{i,j}} \sum_{j\in C} \alpha_j$$

sous contraintes :

$$\forall i \in F, \sum_{j\in C} \beta_{ij} \leq f_i$$

$$\forall i \in F, \forall j \in C, \alpha_j - \beta_{ij} \leq c_{ij}$$

$$\forall i \in F, \forall j \in C, \beta_{ij} \geq 0$$

$$\forall j \in C, \alpha_j \geq 0$$

2.4 Recherche d'une solution

On suppose que l'on a une solution du problème primal. On a donc un sous-ensemble $I \subseteq F$ tel que $y_i = 1 \iff i \in I$ et une fonction $\Phi : C \to I$ telle que $x_{ij} = 1 \iff \Phi(j) = i$.

2.4.1 Conditions des écarts-complémentaires

On note (α, β) une solution optimale pour le problème dual. On suppose qu'il existe une solution admissible, et comme le primal et le dual sont des problèmes linéaires, par théorème de dualité forte, on a $Val(Primal) = Val(Dual)$. On écrit donc les conditions des écarts-complémentaires.

On obtient les équations suivantes :

$$y_i(f_i - \sum_{j\in C} \beta_{ij}) = 0 \quad \forall i \in F$$

$$x_{ij}(c_{ij} + \beta_{ij} - \alpha_j) = 0 \quad \forall i \in F,\ \forall j \in C$$

$$\alpha_j(1 - \sum_{i\in F} x_{ij}) = 0 \quad \forall j \in C$$

$$\beta_{ij}(x_{ij} - y_i) = 0 \quad \forall i \in F, \forall j \in C$$

On étudie la première équation. On sait que $\forall i \in I, y_i = 1$ car toutes les installations du sous-ensemble I sont ouvertes. On en déduit que $\forall i \in I, f_i = \sum_{j \in C} \beta_{ij}$. Or, $\beta_{ij} = 0$ si la ville j n'est pas connectée à l'installation i. On en déduit que $\forall i \in I, f_i = \sum_{j|\Phi(j)=i} \beta_{ij}$. Cela s'interprète en disant que le coût d'ouverture de chaque installation ouverte est entièrement payé. On peut aussi penser en β_{ij} comme étant le montant que la ville j à payé pour ouvrir l'installation i.

On suppose que $\Phi(j) = i$. Cela signifie que la ville j est connectée à l'installation i. Elle ne contribuera donc pas à l'ouverture d'autres installations. On en déduit que $\forall i' \neq i, \beta_{i'j} = 0$. De plus, comme $x_{ij} = 1$, on a $c_{ij} = \alpha_j - \beta_{ij}$. On peut donc interpréter α_j comme le prix total payé par la ville j, β_{ij} comme la part de ce coût contribuant à l'ouverture de l'installation i et c_{ij} comme la part allant à l'utilisation de l'arc $i - j$.

2.4.2 Relaxation des conditions des écarts complémentaires

On suppose que l'on relâche les conditions des écarts complémentaires issus du primal, mais pas celles issues du dual. On écrit donc :

$$\forall j \in C, \frac{1}{3} c_{\Phi(j)j} \leq \alpha_j - \beta_{\Phi(j)j} \leq c_{\Phi(j)j}$$

$$\forall i \in I, \frac{1}{3} f_i \leq \sum_{j|\Phi(j)=i} \beta_{ij} \leq f_i$$

On obtient alors un algorithme avec un facteur d'approximation 3. Mais on souhaiterait obtenir un meilleur facteur d'approximation.

Pour cela, on introduit deux types de villes :

- Ville directement connectée : elles sont les seules à payer pour l'ouverture des installations. On ne relâche pas les conditions primales. On a toujours : $x_{ij}(c_{ij} + \beta_{ij} - \alpha_j) = 0$ et $y_i(f_i - \sum_{j \in C} \beta_{ij}) = 0$.
- Ville indirectement connectée : elles ne paient pas pour l'ouverture des installations. Ainsi, $\beta_{ij} = 0$ toujours pour les villes indirectement connectées. On ne relâche qu'une condition primale. On a désormais : $y_i(f_i - \sum_{j \in C} \beta_{ij}) = 0)$ et $\frac{1}{3} c_{\Phi(j)j} \leq \alpha_j \leq c_{\Phi(j)j}$.

3 Algorithme

A l'aide de la relaxation présentée dans le chapitre 2, nous cherchons à construire un algorithme fournissant une solution approchée du problème basée sur l'article [1]. L'idée principale de l'algorithme est que chaque ville j continue à augmenter sa variable duale, α_j, jusqu'à ce qu'elle soit connectée à une installation ouverte. Toutes les autres variables primaires et duales suivent ce changement, en essayant de satisfaire les conditions d'écarts complémentaires.

3.1 Description de l'algorithme

L'algorithme est constitué de deux phases. Lors de la première, il cherche une solution faisable du dual et détermine un ensemble d'arêtes "spéciales", qui sont des arêtes entre les villes et les installations qu'elles contribuent à l'ouverture (installations qu'elles aident à payer, cela signifie que (i, j) est une arête spéciale si $\beta_{i,j} > 0$), et un ensemble d'installations temporairement ouvertes (installations i qu'ont été totalement payé par les villes, c'est-à-dire $\sum_{j \in C} \beta_{i,j} = f_i$, dans la fin de la phase 1). La seconde phase consiste à déterminer un sous-ensemble d'installations à ouvrir parmi celles temporairement ouvertes et une fonction reliant chaque ville à une installation ouverte. On va décrire les deux phases en détaille ci dessous, et après on fait une autre description de l'algorithme avec quelques modifications qu'on a ajouté pour qu'il marche pour plus d'instances différents.

Il est très important de mentionner que notre algorithme ne fonctionne que pour le cas métrique, c'est-à-dire le cas où le coût de liaison des installations avec les villes, $c_{i,j}$, satisfait à l'inégalité triangulaire

$$c_{i,j} \leq c_{i,j'} + c_{i',j'} + c_{i,j'} \qquad \forall i, i' \in F, \forall j, j' \in C$$

Nous allons essayer de mieux comprendre cette restriction dans une autre section de ce chapitre.

3.1.1 Phase 1

On cherche la solution du problème dual la plus grande possible.

Au temps 0, chaque ville est <u>non-connectée</u> : elle n'est connectée à aucune installation. Pour chaque ville j non-connectée, la valeur de α_j augmente de 1 à chaque pas de temps. Lorsque α_j atteint la valeur c_{ij} pour un certain i, l'arête (i, j) est déclarée comme étant serrée. Afin de s'assurer que l'une des contraintes du problème dual ($\forall i \in F, \forall j \in C, \alpha_j - \beta_{ij} \leq c_{ij}$), on augmente β_{ij} dès que $\alpha_j = c_{ij}$. Cela traduit le fait que la ville j paie pour l'ouverture de l'installation i. On dit qu'une arête (i, j) telle que $\beta_{ij} > 0$ est spéciale.

Soit $i \in F$. Si $\sum_{j \in C} \beta_{ij} = f_i$, alors l'installation i est entièrement payée. On dit alors qu'elle est temporairement ouverte. Dès lors, toutes les villes non-connectées ayant une arête serrée entre elles et l'installation i seront désormais déclarées connectées. L'installation i est appelée Témoin de connexion pour toutes les villes qui lui sont reliées. Les variables α_j des villes connectées ne seront plus augmentées.

Si on considère une installation i ouverte et une ville j non connectée. Dès que la ville j obtient une arête serrée avec l'installation i, elle est déclarée connectée et son témoin de connexion est i. En revanche, comme on a déjà $\sum_{j' \in C} \beta_{ij'} = f_i$, on ne modifie pas β_{ij}, qui vaut toujours 0. L'arête (i, j) n'est donc pas spéciale.

La phase 1 termine dès que toutes les villes sont connectées.

Remarque : il est possible qu'une ville aie payé pour l'ouverture de plusieurs installations. Elle ne peut être connectée qu'à une. Ce problème est traité dans la phase 2 de l'algorithme. Toutefois, nous n'avons pas réussi à trouver un exemple de taille petite où la phase 2 était nécessaire pour trouver la solution optimale. Dans l'exemple que nous allons décrire ci dessous, nous allons voir que les témoins de connexions trouvé dans la phase 1 donnent déjà la solution optimale du problème.

3.1.2 Phase 2

Tout d'abord, on introduit plusieurs notations :

- F_t : l'ensemble des installations temporairement ouvertes
- T : le sous-graphe de G constitué des arêtes spéciales
- T^2 : le sous-graphe de G constitué des arêtes (u, v) telles qu'il existe un chemin de longueur au plus 2 entre u et v dans T
- H : le sous-graphe de T^2 induit par F_t
- I : un sous-ensemble indépendant maximal de H

Les installations de I sont déclarées ouvertes.

Soit une ville j. On définit $\mathcal{F}_j = \{i \in F_t | (i, j)$ est spécial$\}$. Comme I est un ensemble indépendant, au plus une installation de $\mathcal{F}_j$ est ouverte. On a alors plusieurs cas :

- Il existe une installation $i \in \mathcal{F}_j$ qui est ouverte. Alors, on pose $\Phi(j) = i$ et on déclare la ville j directement connectée.
- On considère l'arête serrée (i', j) telle que i' est le témoin de connexion de j.
  - Si $i' \in I$, alors on pose $\Phi(j) = i'$ et on déclare la ville i' directement connectée. On note que, dans ce cas, $\beta_{i'j} = 0$.
  - Sinon, soit k un voisin de i' dans le graphe H, tel que $k \in I$. Dans ce cas, on pose $\Phi(j) = k$ et on déclare la ville j indirectement connectée.

I et Φ définissent une solution entière du primal : $x_{ij} = 1\Phi(j) = i$ et $y_i = 1i \in I$.

Les valeurs de α_j et β_{ij} obtenues à la fin de la phase 1 forment une solution réalisable du problème dual.

3.2 Analyse de l'algorithme

3.2.1 Paiement des coûts d'ouverture des installations

On cherche à interpréter le paiement des coûts d'ouverture des installations et de connexion des installations aux villes. Pour cela, on pose α_j^f la contribution de la ville j à l'ouverture des installations et α_j^e la contribution de la ville j à la connexion à une installation. On a alors $\alpha_j = \alpha_j^f + \alpha_j^e$.

- Si la ville j est indirectement connectée, alors $\alpha_j^f = 0$ et $\alpha_j = \alpha_j^e$.
- Si la ville j est directement connectée, alors, $\alpha_j = c_{ij} + \beta_{ij}$, en notant $i = \Phi(j)$. on a alors $\alpha_j^f = \beta_{ij}$ et $\alpha_j^e = c_{ij}$.

On va tout d'abord prouver que, pour $i \in I$, $\sum_{j|(i,j) \text{ special}} \alpha_j^f = f_i$. En effet, on a $\sum_{j|(i,j) \text{ special}} \beta_{ij} = f_i$ par définition d'une arête spéciale et car l'installation i est ouverte, donc temporairement ouverte à la fin de la phase 1. Par indépendance de I, toute ville ayant contribué à l'ouverture de i y est directement connectée. Pour ces villes, on a donc $\alpha_j^f = \beta_{ij}$. Les autres doivent satisfaire $\alpha_j^f = 0$. On en déduit que $\sum_{j|(i,j) \text{ special}} \alpha_j^f = f_i$. Ainsi, seules les villes directement connectées paient pour l'ouverture des installations. De cette équation, on déduit que $\sum_{i \in I} f_i = \sum_{j \in C} \alpha_j^f$ car, pour une ville indirectement connectée, $\alpha_j^f = 0$.

On montre également que pour une ville indirectement connectée j telle que $\Phi(j) = i$, on a $c_{ij} \leq 3\alpha_j^e$. Pour cela, on considère i' le témoin de connexion de j à i. j étant indirectement connectée à i, (i, i') est une arête dans H. Il existe donc une ville j' telle que (i, j') et (i', j') sont des arêtes spéciales. On note t_1 le temps auquel l'installation i a été ouverte lors de la phase 1, et t_2 ce temps pour l'installation i'. De plus, l'arête (i', j) est serrée. On en déduit que $\alpha_j \geq c_{i'j}$. De même, les arêtes (i', j') et (i, j') sont serrées. Donc $\alpha_{j'} \geq c_{i'j'}$ et $\alpha_{j'} \geq c_{ij'}$. Or, d'après le déroulement de la phase 1, $\alpha_{j'}$ cesse de croître dès lors qu'une installation avec laquelle la ville j' a une arête serrée est ouverte. Donc $\alpha_{j'} \leq \min(t_1, t_2)$. Par ailleurs, i' est le témoin de connexion pour j, donc $\alpha_j \geq t_2$. On en déduit que $\alpha_{j'} \leq \alpha_j$. Par inégalité triangulaire, on a $c_{ij} \leq c_{ij'} + c_{i'j'} + c_{i'j} \leq 3\alpha_j = 3\alpha_j^e$ car la ville j est indirectement connectée.

Pour une ville directement connectée, on a $c_{ij} = \alpha_j^e \leq 3\alpha_j^e$. Donc avec l'inégalité prouvée précédemment, on a $\sum_{i \in F, j \in C} c_{ij} x_{ij} \leq 3 \sum_{j \in C} \alpha_j^e$. On sait également que $\sum_{i \in I} f_i \leq \sum_{j \in C} \alpha_j^f$. On obtient finalement l'inégalité suivante :

$$\sum_{i \in F, j \in C} c_{ij} x_{ij} + 3 \sum_{i \in F} y_i f_i \leq 3 \sum_{j \in C} \alpha_j.$$

3.2.2 Estimation du temps d'exécution de l'algorithme

On peut prouver que l'algorithme présenté donne un facteur d'approximation 3 pour le problème présenté en un temps $O(mlogm)$ où $m = n_c \times n_f$ avec n_c le nombre de villes et n_f le nombre d'installations. Pour cela, on trie toutes les arêtes par ordre de coût croissant. Cela nous donne à la fois le moment où une arête devient serrée et l'ordre dans lequel elles deviennent serrées.

3.3 Modification de l'algorithme pour des instances non entières

Nous avons observé que la phase 1 de l'algorithme approximatif, de la façon dont elle est décrit dans l'article, présente un petit problème : l'algorithme augmente les valeurs des α_j pour chaque ville j de 1 jusqu'à que α_j soit égal à un coût $c_{(i,j)}$ pour une installation i. Toutefois, cela restreint notre algorithme à fonctionner seulement avec des contraintes de coûts entières. Pour contourner ce problème, nous avons fait quelques modifications que nous allons décrire dans cette section.

Pour commencer, nous ordonnons les coûts de liaison de chaque ville $j \in C$ avec les installations $i \in F$ dans l'ordre croissant, où C est l'ensemble des noeuds de villes et F est l'ensemble des noeuds des installations. Alors, si on a n villes dans le problème, nous allons avoir n listes de coûts dans l'ordre croissant.

A chaque tour, l'algorithme analyse ce qui se passe avec chaque ville j. Comme décrit dans l'article, le but est d'augmenter les alphas le plus possible sans dépasser les contraintes du problème dual :

$$\alpha_j - \beta_{(i,j)} \leq c_{(i,j)} \quad \forall i \in F,\ \forall j \in C$$

$$\sum_{j \in C} \beta_{(i,j)} \leq f_i$$

où f_i est le prix d'ouverture de l'installation i. De cette façon, on essaie de trouver une solution réalisable du dual, la plus proche possible de la solution optimale.

Toutefois, au lieu d'augmenter chaque α_j de 1 à chaque tour, on commence en les posant égaux aux plus petits coûts (premier élément de la liste des coûts de la ville j dans l'ordre croissant), et on essaie à chaque itération de les augmenter pour être égaux au prochain coût de la liste correspondant à la ville j.

Supposons qu'à l'itération actuelle, on est en train d'analyser la ville j. On suppose que $\alpha_j = c_{(i,j)}$ pour une installation $i \in F$, et que le prochain plus petit coût de la liste est $c_{(i',j)}$ pour l'installation $i' \in F$. Pour savoir si on peut poser $\alpha_j = c_{(i',j)}$, c'est-à-dire, augmenter α_j de $c_{(i',j)} - c_{(i,j)}$, il faut :

1. Vérifier si la ville j n'est pas encore connectée (si elle est connectée, on n'augmente plus son alpha) ;

2. Vérifier si pour tout $i \in F$ qui fait une *arête spéciale* avec j (où la notion d'*arête spéciale* est la même que celle donnée dans l'article) - c'est-à-dire, qui contribue à payer l'ouverture de la facility i - on peut augmenter $\beta_{(i,j)}$.
   On veut maintenir l'inégalité $\alpha_j - \beta_{(i,j)} \leq c_{(i,j)}$ pour tout $i \in F$. Alors, si α_j a déjà dépassé le coût $c_{(i,j)}$ pour un certain i, c'est-à-dire que $\beta_{i,j}$ est une *arête spéciale*, et si on veut à nouveau augmenter α_j, il faut aussi augmenter $\beta_{(i,j)}$ pour maintenir l'inégalité. Mais pour augmenter $\beta_{(i,j)}$, il faut vérifier que l'on ne va pas dépasser l'inégalité d'ouverture de l'installation : $\sum_j \beta_{i,j} \leq f_i$.

Alors la question est, comment on peut vérifier cette deuxième condition - peut-on augmenter le β ?

Pour tout $i \in F$ tel que (i, j) est une *arête spéciale* on vérifie si

$$f_i - \sum_j \beta_{i,j} \geq c_{i',j} - c_{i,j}.$$

- Si oui, alors on peut augmenter $\beta_{(i,j)}$ pour toute installation i où (i, j) est une *arête spéciale*, et donc on peut aussi augmenter α_j. Comme on aura $\alpha_j = c_{(i',j)}$, dans le futur, si on veut augmenter α_j encore, il va falloir aussi augmenter $\beta_{(i',j)}$, alors (i', j) devient lui aussi une *arête spéciale*.
  Si dans ce cas il y a un $i \in F$ pour lequel on a une égalité dans l'expression ci dessus, après avoir augmenté le β correspondant, on aura que cette installation sera totalement payée, et donc elle sera temporairement ouverte. Toutes les villes qui ont contribué à son ouverture seront connectées à elle.
- Sinon, il y a un ensemble I d'installations où $\forall i \in I$ tel que (i, j) est une *arête spéciale*, on aura
  $$f_i - \sum_j \beta_{i,j} < c_{i',j} - c_{i,j}$$
  donc pour ces i on ne peut pas augmenter $\beta_{i,j}$ de $c_{i',j} - c_{i,j}$ et si on ne les augmente pas, on ne peut pas non plus augmenter α_j de cette quantité sans dépasser la première inégalité du Dual.

  Pour résoudre ce problème, on considère

  $$d = \min_{i \in I, (i,j) \textit{arête spéciale}} f_i - \sum_j \beta_{i,j}$$

  On sait donc qu'on peut augmenter les $\beta_{i,j}$ de d, et donc on peut augmenter le α_j de d aussi.

  On peut remarquer que, dans ce cas, on va saturer la contrainte $\sum_j \beta_{i,j} \leq f_i$ pour le i correspondant au minimum d. Alors cette installation sera totalement payée, et donc, selon l'article, on peut la déclarer temporairement ouverte, et on peut connecter la ville j à elle ainsi que toutes les autres villes qui ont contribué à son ouverture.

Dans le cas où la ville j n'a pas encore d'*arête spéciale* avec une autre installation, alors on peut augmenter le α jusqu'au coût $c_{i',j}$, et, alors, (i',j) devient une *arête spéciale*.

Si nous sommes passés par tous les coûts et qu'il y a une ville qui n'a pas encore été connectée, nous continuons à essayer de voir si nous pouvons ou non monter alpha (d'un certain valeur arbitraire) en prenant soin des inégalités de la même manière décrite ci-dessus. Cela peut être le cas par exemple lorsque nous n'avons qu'une seule installation pour plusieurs villes. Dans le premier tour, nous rendrons les α_j égaux au coût de la liaison de chaque ville $j \in C$ à l'unique installation du problème. Mais tous les bêtas seront encore à zéro à la fin de la première tournée, donc l'installation ne sera pas encore temporairement ouverte. Nous devons donc continuer à augmenter alpha pour pouvoir augmenter bêta et payer alors l'installation. Comme toutes les villes auront une arête serré avec cette installation, l'algorithme peut relier toutes les villes à cette installation (la réponse triviale à cette instance).

3.4 Algorithme détaillé dans un exemple (phase 1)

On détaille comment la phase 1 de l'algorithme modifié pour des instances non entières fonctionne dans un petit exemple décrit dans le tableau ci dessous :

Installations						
	$i=1$	$i=2$	$i=3$	$i=4$	$i=5$	$i=6$
Open costs	1	1.5	0.5	1	1	1
$j=1$	4	2	6.5	7	4	4
$j=2$	4.5	6.5	3.5	2	3	2.5
$j=3$	2.5	5.5	6.5	6	3	3.5
$j=4$	6	6.5	0.5	1	4	3

Chaque tour de l'algorithme correspond à un parcours de toutes les villes (dans la tables représentées par les js). A la fin de la description de chaque tour, s'il y a eu des changements dans ce tour, il y a aussi deux tableaux : un qui indique plus clairement les valeurs des α pour chaque ville, si elle était connectée et à quelle installation ; l'autre qui indique plus clairement les arêtes qui ont devenus *arêtes spéciales*, et le valeur des β pour elles (on se souvient que les β pour des arêtes qui ne sont pas des *arêtes spéciales* sont tous 0).

Tour 1 **:**

Les plus petit coûts de liaison d'une installation à une ville pour chaque ville sont donnés par :

j = 1	j = 2	j = 3	j = 4
$c_{(2,1)}$	$c_{(4,2)}$	$c_{(1,3)}$	$c_{(3,4)}$
2	2	2.5	0.5

- Pour commencer, l'algorithme fait α_j pour toute ville $j \in \{1, 2, 3, 4\}$ être égal au plus petit coût lié à cette ville.
- Comme pour la ville $j = 1$, $\alpha_1 = c_{(2,1)}$, dans le futur si on veut augmenter α_1 il va falloir aussi augmenter $\beta_{(2,1)}$ pour maintenir la contrainte du Dual $\alpha_j - \beta_{(i,j)} \leq c_{(i,j)}$. Alors on va dire que $(2, 1)$ est une *arête spéciale*.
- De la même façon, si on veut augmenter dans le futur α_2, α_3, ou α_4, il va falloir augmenter $\beta_{(4,2)}$, $\beta_{(1,3)}$ ou $\beta_{(3,4)}$ (respectivement pour chaque ville). Alors $(4, 2)$, $(1, 3)$, et $(3, 4)$ deviennent des *arêtes spéciales*.

Ville	Alpha	Connectée	*Témoin de connexion*
1	2	Non	None
2	2	Non	None
3	2.5	Non	None
4	0.5	Non	None

Arête spéciale	Beta
$(2, 1)$	0
$(4, 2)$	0
$(1, 3)$	0
$(3, 4)$	0

Tour 2 :
Les deuxièmes plus petits coûts pour chaque ville sont :

j = 1	j = 2	j = 3	j = 4
$c_{(1,1)}$	$c_{(6,2)}$	$c_{(5,3)}$	$c_{(4,4)}$
4	2.5	3	1

On décrit ce qui va se passer pour chaque ville :

- Pour la ville 1, on regarde si on peut augmenter $\alpha_1 = 2$ jusqu'à la valeur du prochain coût $c_{(1,1)} = 4$. Pour cela il faut sommer 2 à α_1, mais pour que cela soit possible, il faut pouvoir sommer 2 à sa seule *arête spéciale* $\beta_{(2,1)}$.
  Comme le prix d'ouverture de l'installation 2 est donnée par 1.5, pour ne pas dépasser la deuxième contrainte du dual, on peut augmenter $\beta_{(2,1)}$ que de 1.5. Alors on fait $\alpha_1 = 3.5$, et $\beta_{(2,1)} = 1.5$.
  Dans ce cas, $f_2 = \beta_{(2,1)}$ et donc l'installation 2 sera totalement payée. On la déclare donc temporairement ouverte, et comme la ville 1 a contribué pour cela, on connecte la ville 1 à l'installation 2.
- Pour la ville 2, on veut augmenter $\alpha_2 = 2$ de 0.5 pour qu'il soit égal au prochain coût $c_{(6,2)} = 2.5$. Il faut donc augmenter $\beta_{(4,2)}$ de 0.5 aussi. Comme $f_4 = 1$, on peut faire cela facilement sans dépasser aucune contrainte. Alor,s on a dans ce tour $\alpha_2 = 2.5$ et $\beta_{(4,2)} = 0.5$. Il faut maintenant 0.5 pour finir de payer l'installation 4. Comme $\alpha_2 = c_{(6,2)}$ à la fin de ce tour, $(6, 2)$ va devenir une *arête spéciale* aussi.

- Pour la ville 3, $\alpha_3 = 2.5$ et on veut l'augmenter jusqu'à $c_{(5,3)} = 3$. Pour le faire il faut augmenter $\beta_{(1,3)}$ de 0.5. Comme $f_1 = 1$, $\beta_{(1,j)} = 0 \; \forall j \neq 1$, on peut faire cela sans dépasser aucune contrainte. On a donc $\alpha_3 = 3$, et $\beta_{(1,3)} = 0.5$. Il faut encore 0.5 si on veut ouvrir l'installation 1.
  Dans ce cas, on a $\alpha_3 = c_{(5,3)}$, et donc $(5, 3)$ devient elle aussi une *arête spéciale*.
- Finalement, pour la ville 4 on veut que $\alpha_4 = 0.5$ devenienne $c_{(4,4)} = 1$. Pour cela il faut pouvoir augmenter $\beta_{(3,4)}$ de 0.5. Comme $f_3 = 0.5$, et tous les autres β liés à cette installation sont égaux à 0, on peut faire cela sans dépasser la contrainte du dual. On aura donc $\alpha_4 = 1$, et $\beta_{(3,4)} = 0.5$.
  Dans ce cas, on va avoir $f_3 = \beta_{(3,4)}$, donc l'installation 3 est totalement payée, et alors elle est déclarée temporairement ouverte. Comme la ville 4 a aidé, on connecte la ville 4 à l'installation 3.
  Comme la ville 4 est maintenant connectée, on ne va plus augmenter α_4 dans le prochain tour, et alors $\alpha_4 = c_{(4,4)}$, $(4, 4)$ ne devient pas une *arête spéciale*.

Ville	Alpha	Connectée	*Témoin de connexion*
1	3.5	Vrai	2
2	2.5	Faux	None
3	3	Faux	None
4	1	Vrai	3

Arête spéciale	Beta
$(2, 1)$	1.5
$(4, 2)$	0.5
$(1, 3)$	0.5
$(3, 4)$	0.5
$(6, 2)$	0
$(5, 3)$	0

Tour 3 **:**

Les troisièmes plus petits coûts pour chaque ville qui ne sont pas encore connectées sont :

j = 2	j = 3
$c_{(5,2)}$	$c_{(6,3)}$
3	3.5

- Pour la ville 2, on veut augmenter $\alpha_2 = 2.5$ de 0.5 pour devenir égal à $c_{(5,2)} = 3$. Les seules *arêtes spéciales* liées à la ville 2 sont $(6, 2)$ et $(4, 2)$, il faut donc pouvoir augmenter $\beta_(4, 2) = 0.5$ et $\beta_{(6,2)} = 0$ de 0.5.
  Comme le prix d'ouverture de l'installation 6 est 1, et comme il n'y a pas d'autres *arêtes spéciales* avec cette installation, on peut poser $\beta_{(6,2)} = 0.5$.

Comme le prix d'ouverture de l'installation 4 est 1, il n'y a pas d'autres *arêtes spéciales* avec cette installation, et $f_4 - \beta_{(4,2)} = 1 - 0.5 = 0.5$, on peut faire $\beta_{(4,2)} = 1$.
Donc, on peut aussi faire $\alpha_2 = 3$.
Dans la fin de ce tour, comme $f_4 = \beta_{(4,2)}$, cette installation va être payée, et alors, temporairement ouverte. Comme la ville 2 a contribué, on connecte la ville 2 à l'installation 4.

- Finalement, pour la ville 3, on a $\alpha_3 = 3$ et on veut l'augmenter de 0.5 pour devenir égal à $c_{(6,3)} = 3.5$. Pour cela il faut augmenter les β liés aux deux *arêtes spéciales* avec cette ville : $(1, 3)$ et $(5, 3)$.
Le prix de $f_1 = 1$, le seul β non nul lié à cette installation est $\beta_{(1,3)} = 0.5$, alors on peut encore augmenter ce β de 0.5.
Le prix de $f_5 = 1$, le seul β non nul lié à cette installation est $\beta_{(5,3)} = 0$, alors on peut encore augmenter ce β de 0.5.
Alors, on peut bien faire $\alpha_3 = 3.5$.
Dans ce cas, pour l'installation 1 on aura $f_1 = \beta_{(1,3)}$, et donc cette installation est totalement payée par la ville 3, alors l'installation 1 est temporairement ouverte et on connecte la ville 3 à elle.

Ville	Alpha	Connecté	*Témoin de connexion*
1	3.5	Vrai	2
2	3	Vrai	4
3	3.5	Vrai	1
4	1	Vrai	3

Arêtes spéciales	Beta
$(2, 1)$	1.5
$(4, 2)$	1
$(1, 3)$	1
$(3, 4)$	0.5
$(6, 2)$	0.5
$(5, 3)$	0.5

Comme toutes les villes ont déjà été connectées, la phase 1 finit ici.

Les résultats sont :

Villes Installations

$$1 \Longrightarrow 2$$

$$2 \Longrightarrow 4$$

$$3 \Longrightarrow 1$$

$$4 \Longrightarrow 3$$

Dans ce cas, seulement avec la phase 1, on obtient le même résultat que l'algorithme exact. On gagne aussi une solution réalisable du Dual, qui dans ce cas est bien la solution optimale.

3.4.1 Détails de la phase 2

Nous n'avons pas trouvé un exemple de taille petite où la phase 1 n'était pas suffisant, alors malheureusement nous ne pouvons pas faire une section en décrivant un exemple de l'algorithme dans la phase 2. Toutefois, nous pouvons faire une description de comment nous avons codé cette phase et de quelques problèmes que nous avons rencontré dans le cas des instances plus grands où la phase 2 était nécessaire.

Même si la phase 2 comporte beaucoup de nouvelles définitions de graphes, l'idée est basique. Le problème de la phase 1 est qu'en théorie, nous pouvons avoir une ville qui contribue à l'ouverture de plus d'une installation (c'est-à-dire une $j \in C$ tel que $\beta_{i,j} > 0$ pour plus d'une installation i). Pour résoudre ce problème, l'idée est de créer d'abord un graphe qui contiendra comme noeuds toutes les installations temporairement ouvertes, et nous lierons deux installations i, i' dans ce graphe seulement s'il y a une ville j telle qu'elle a une arête spécial avec ces deux installations.

C'est exactement la définition de T^2 qui est le sous-graphe de T (le graphe avec seulement les arêtes spéciales) tel que nous avons des arêtes si et seulement s'il y a un chemin de liaison d'au plus 2 entre les noeuds, et avec les noeuds étant seulement ceux des installations temporairement ouvertes. Enfin, choisir un ensemble indépendant maximal de H signifie que nous voulons trouver le plus grand (ou l'un des plus grands) ensemble de noeuds tel que tous les noeuds qui le composent n'ont pas d'arête qui les relie. Comme deux installations n'ont une arête qui les relie que si elles "partagent" une ville comme arête spéciale, l'ensemble indépendant ne comportera que des installations payées par des villes différentes.

Le problème majeur que nous avons trouvé en le codant, est que cet ensemble indépendant est choisi au hasard, et pour les grandes instances, parfois l'algorithme trouve les bonnes installations à ouvrir (les mêmes que la version exacte), parfois non. Nous nous demandons donc s'il n'y a pas une façon plus intelligente de choisir l'ensemble indépendant maximal à ouvrir au lieu de le choisir au hasard.

3.5 Le cas métrique

Dans cette section le but est d'essayer de comprendre pourquoi il faut l'hypothèse que les coûts satisfassent à l'inégalité triangulaire.

Nous savons par le théorème de dualité forte que dans ce problème, si $(x_{i,j})_{i\in F, j\in C}$, $(y_i)_{i\in F}$ était la solution optimal du primal, et $(\alpha_j)_{j\in C}$, $(\beta_{i,j})_{i\in F, j\in C}$ la solution optimale du dual, alors

$$\sum_{i\in F, j\in C} c_{i,j}x_{i,j} + \sum_{i\in F} y_i f_i = \sum_{j\in C} \alpha_j$$

Comme on a vu dans le début du chapitre, la solution de l'algorithme approximatif nous permet d'avoir

$$\sum_{i\in F, j\in C} c_{i,j}x_{i,j} + 3\sum_{i\in F} y_i f_i \leq 3\sum_{j\in C}\alpha_j$$

Où on avait que la solution approximative du dual paye entièrement le coût d'ouverture des installations

$$\sum_{i\in I} f_i = \sum_{j\in C}\alpha_j^f,$$

mais ne paye que $\frac{1}{3}$ du coût de connexion, car

$$\sum_{j\in C} c_{i,j}x_{i,j} \leq 3\sum_{j\in C}\alpha_j^e.$$

Nous avons donc une borne indiquant de combien le résultat de l'algorithme approximatif peut être loin de la solution optimale. Toutefois, pour faire ce calcul, on a utilisé le fait que $c_{i,j} \leq 3\alpha_j^e$.C'est évident dans le cas où la ville est directement connectée (car $c_{i,j} = \alpha_j^3$), mais ce n'est valide pour les villes indirectement connectées que si on a l'inégalité triangulaire des coûts. Alors, dans le cas non métrique, nous n'avons pas ce résultat, et nous ne pouvons pas garantir que l'algorithme approximatif donne une approximation par un facteur de 3.

Nous pouvons aussi essayer de comprendre ce qui se passe avec un exemple. Supposons que nous avons l'instance ci-dessous.

Installations						
	$i=1$	$i=2$	$i=3$	$i=4$	$i=5$	$i=6$
Open costs	2	1.75	4.5	11.5	7.75	12.5
$j=1$	2.75	4.25	0.25	1.75	2.25	2
$j=2$	2	2.5	4	5	1.5	1.25
$j=3$	4.75	2	1.5	1.75	0.75	3.5
$j=4$	0.5	0.75	0.75	3.25	0.5	2.25

Les coûts de cette instance ne satisfont pas l'inégalité triangulaire. Par exemple : $c_{1,3} = 4.75$, $c_{1,4} = 0.5$, $c_{5,4} = 0.5$ et $c_{5,3} = 0.75$. Alors, $c_{1,4} + c_{5,4} + c_{5,3} = 1.75 < c_{1,3}$.

Nous avons que la solution exacte est donné par

Villes Installations

$$1 \Longrightarrow 3$$

$$2 \Longrightarrow 1$$

$$3 \Longrightarrow 3$$

$$4 \Longrightarrow 1$$

et le coût est 10.75.

Toutefois, l'algorithme approximatif donne après la phase 1

Ville	Alpha	*Témoin de connexion*
1	2.5	3
2	2.25	1
3	2.25	2
4	2.25	1

La phase 1 ne donne pas un résultat dans la borne prévue par l'algorithme :

— La somme des coûts vaut : $c_{3,1} + c_{1,2} + c_{2,3} + c_{1,4} = 0.25 + 2 + 2 + 0.5 = 4.75$;

— La somme des alpha vaut : $\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 = 3 * 2.25 + 2.5 = 9.25$;

— La somme des coûts des installations ouvertes vaut : $f_1 + f_2 + f_3 = 2 + 1.75 + 4.5 = 8.25$

Alors, somme des coûts + $3*$ somme des coûts des installations = $4.75 + 3 * 8.25 = 29.5$, alors que $3*$ somme des alpha = $3 * 9.25 = 27.75$. Comme $27.75 < 29.5$, on n'a pas le résultat qu'on voulait.

Si on continue avec la phase 2, comme la phase 1 trouve que parmi les arêtes spéciales il y a $(1, 4)$, $(2, 4)$ et $(3, 4)$, alors dans le graphe H on va avoir les arêtes $(1, 2)$, $(2, 3)$ et $(1, 3)$, donc l'algorithme va ouvrir seulement une installation parmi 1, 2, et 3. Cela ne donne pas du tout le résultat optimal (même si maintenant on est dans les bornes du résultat précédent).

Alors, même si on ne comprend pas encore entièrement où dans les calculs de l'algorithme l'inégalité triangulaire pour les coûts est nécessaire, on arrive à voir que, dans certains cas où les coûts ne sont pas dans le cas métrique, le résultat obtenu diffère du résultat optimal.

3.6 Problème des k-médianes

Une autre modélisation pour le problème de placement est le problème des k-médianes. Il est très similaire au problème de localisation des installations, avec deux différences majeures : il n'y a pas de coût pour l'ouverture des installations, et nous avons une limite supérieure k au nombre d'installations qui peuvent être ouvertes. Cette limite k fait partie des données de départ du problème. La modélisation peut être écrite de la façon suivante (x_{ij} et y_i ont la même signification que précédemment) :

$$\min_{x_{ij}, y_i} \sum_{i \in F, j \in C} c_{ij} x_{ij}$$

sous contraintes :

$$\forall j \in C, \sum_{i \in F} x_{ij} \geq 1$$

$$\forall i \in F, \forall j \in C, y_i - x_{ij} \geq 0$$

$$\sum_{i \in F} y_i \leq k$$

$$\forall i \in F, \forall j \in C, x_{ij} \in \{0, 1\}$$

$$\forall i \in F, y_i \in \{0,1\}$$

On écrit la relaxation continue du problème :

$$\min_{x_{ij},y_i} \sum_{i\in F, j\in C} c_{ij}x_{ij}$$

sous contraintes :

$$\forall j \in C, \sum_{i\in F} x_{ij} \geq 1$$

$$\forall i \in F, \forall j \in C, y_i - x_{ij} \geq 0$$

$$\sum_{i\in F} y_i \leq k$$

$$\forall i \in F, \forall j \in C, x_{ij} \geq 0$$

$$\forall i \in F, y_i \geq 0$$

Le dual du problème s'écrit quant à lui :

$$\max \sum_{j\in C} \alpha_j - zk$$

sous contraintes :

$$\forall i \in F, \forall j \in C, \alpha_j - \beta_{ij} \leq c_{ij}$$

$$\forall i \in F, \sum_{j\in C} \beta_{ij} \leq z$$

$$\forall j \in C, \alpha_j \geq 0$$

$$\forall i \in F, \forall j \in C, \beta_{ij} \geq 0$$

$$z \geq 0$$

z est le multiplicateur de Lagrange associé à la contrainte d'avoir au plus k installations ouvertes.

On remarque toutefois que la forme du problème des k-médianes est très similaire à celle du problème étudié précédemment. Considérons une instance du premier problème. On pose, $\forall i \in F, f_i = z$, où z est une constante. On obtient des solutions optimales du premier problème ainsi posé, que l'on note (x, y) et (α, β).

Le théorème de dualité forte donne alors $\sum_{i\in F, j\in C} c_{ij}x_{ij} + \sum_{i\in F} zy_i = \sum_{j\in C} \alpha_j$. Si l'on suppose qu'exactement k installations sont ouvertes, alors (x, y) et (α, β, z) sont solutions optimales respectivement du primal et du dual du problème des k-médianes.

Nous pouvons donc utiliser l'algorithme approximatif que nous avions pour résoudre le problème de localisation des installations pour construire un algorithme approximatif pour le problème des k-médianes.

Malheureusement, nous n'avons pas eu le temps de coder et de mieux comprendre cette partie de l'article. Si nous devions continuer à travailler sur ce sujet, nous pourrions travailler davantage sur ce problème.

4 Conclusion

Le but de ce projet était d'en apprendre plus sur le problème de localisation d'entrepôts, et le problème des k-médianes, deux types de problèmes linéaires en nombres entiers de grande importance aujourd'hui, en particulier lorsqu'il s'agit du problème de placement du réseau 5G. En particulier, nous voulions mieux comprendre l'algorithme approximatif proposé par l'article [1] pour le problème de localisation des installations, et comment le lier au problème des k-médianes.

Avec l'aide des outils appris pendant le cours de Recherche Opérationnelle [2], nous avons pu refaire les calculs de l'article, et en codant l'algorithme en Python et en le comparant avec l'algorithme exact pour le même problème (aussi codé en Python), nous avons pu mieux comprendre comment l'algorithme fonctionne. De plus, nous avons vu les limites de la version de l'algorithme proposée dans l'article, en ne permettant que les instances entières, et nous avons modifié pour une nouvelle version qui pourrait accepter n'importe quel type d'instance.

Bien que nous ayons pu mieux décrire le schéma Primal-Dual utilisé par l'article, il reste encore du travail à faire pour mieux comprendre comment savoir si une instance donnée doit passer par la phase 2 du code ou non. De même, même si nous observons la nécessité d'avoir une instance où les coûts satisfont l'inégalité triangulaire afin de répondre aux résultats théoriques de l'article, il n'est toujours pas clair comment cela affecte réellement la procédure de l'algorithme.

Enfin, pour un travail ultérieur, une tentative de codage de l'algorithme qui fonctionne pour le problème des k-médianes en utilisant celui déjà codé pour le problème de localisation des installations comme proposé dans la fin du chapitre 3 pourrait être faite, ce qui clarifierait probablement aussi les procédures théoriques décrites dans l'article [1]. D'autres articles qui traitent du problème des k-médianes pourraient aussi apporter une nouvelle vision du problème, comme dans [3]. Comme ce dernier problème est lié à celui du clustering, qui joue un rôle central dans de nombreux domaines, y compris la data science et le machine learning, le comprendre en profondeur, et comprendre les algorithmes approximatifs pour le résoudre, est très intéressant en pratique et a de multiples applications.

Bibliographie

[1] Kamal Jain et Vijay V. Vazirani : Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and lagrangian relaxation. *journal of the ACM*, 48, 2001.

[2] J. Frédéric Bonnans et Stéphane Gaubert : *Recherche Opérationnelle : Aspects Mathématiques et Applications.* 2020.

[3] Sara Ahmadian, Ashkan Norouzi-Fard, Ola Svensson et Justin Ward : Better guarantees for k-means and euclidean k-median by primal-dual algorithms. *CoRR*, abs/1612.07925, 2016.